VLDB2011
37th International Conference on Very Large Data Bases
Aug 29 - Sep 3: The Westin, Seattle, WA

Efficient Diversification of Web Search Results

G. Capannini, F. M. Nardini, R. Perego, and F. Silvestri
ISTI-CNR, Pisa, Italy

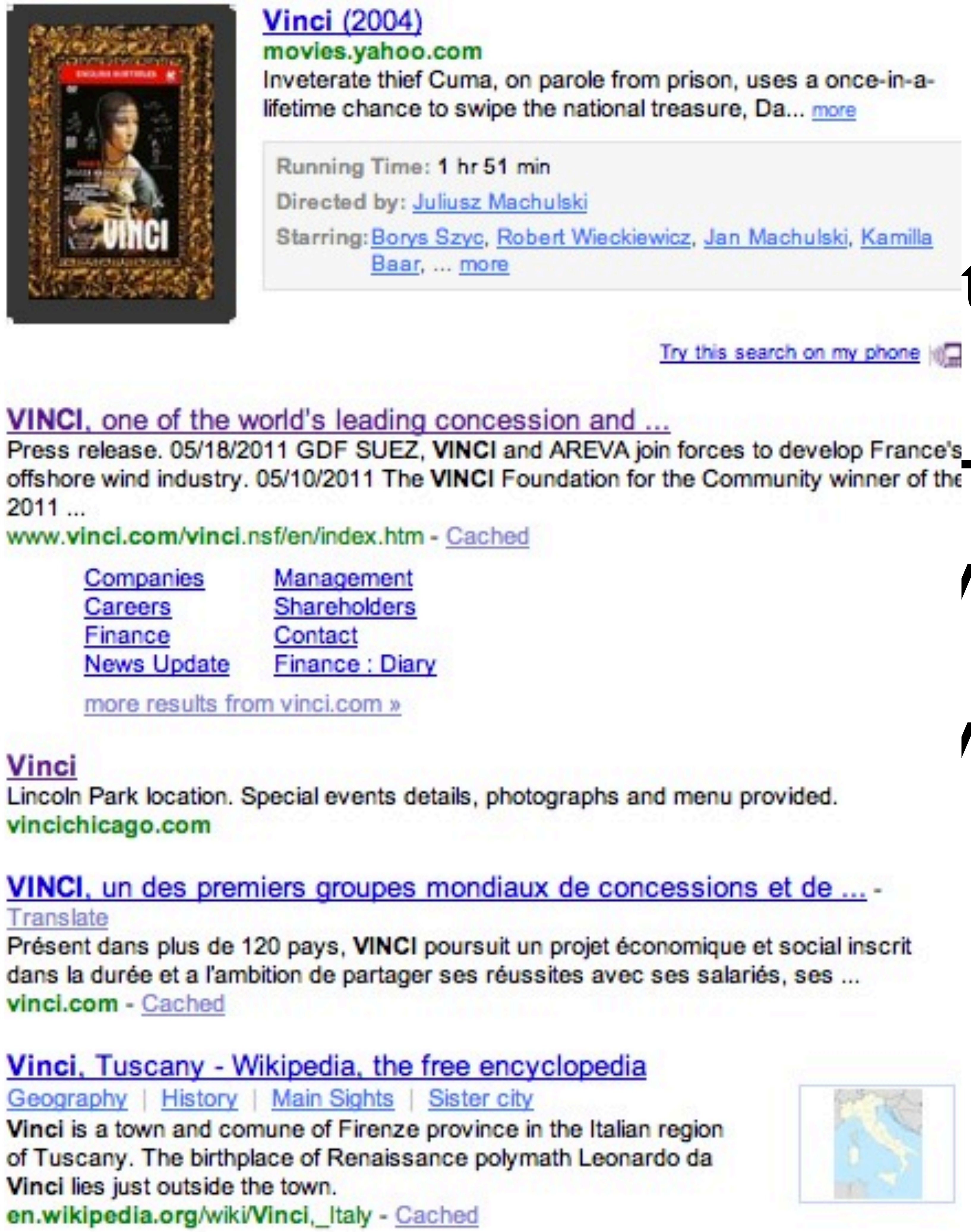
ISTI

HPC Laboratory

Web Search Results Diversification

- Query: "Vinci", what is the user's intent?
  - Information on Leonardo da Vinci?
  - Information on Vinci, the small village in Tuscany?
  - Information on Vinci, the company?
  - Others?

Web Search Results Diversification

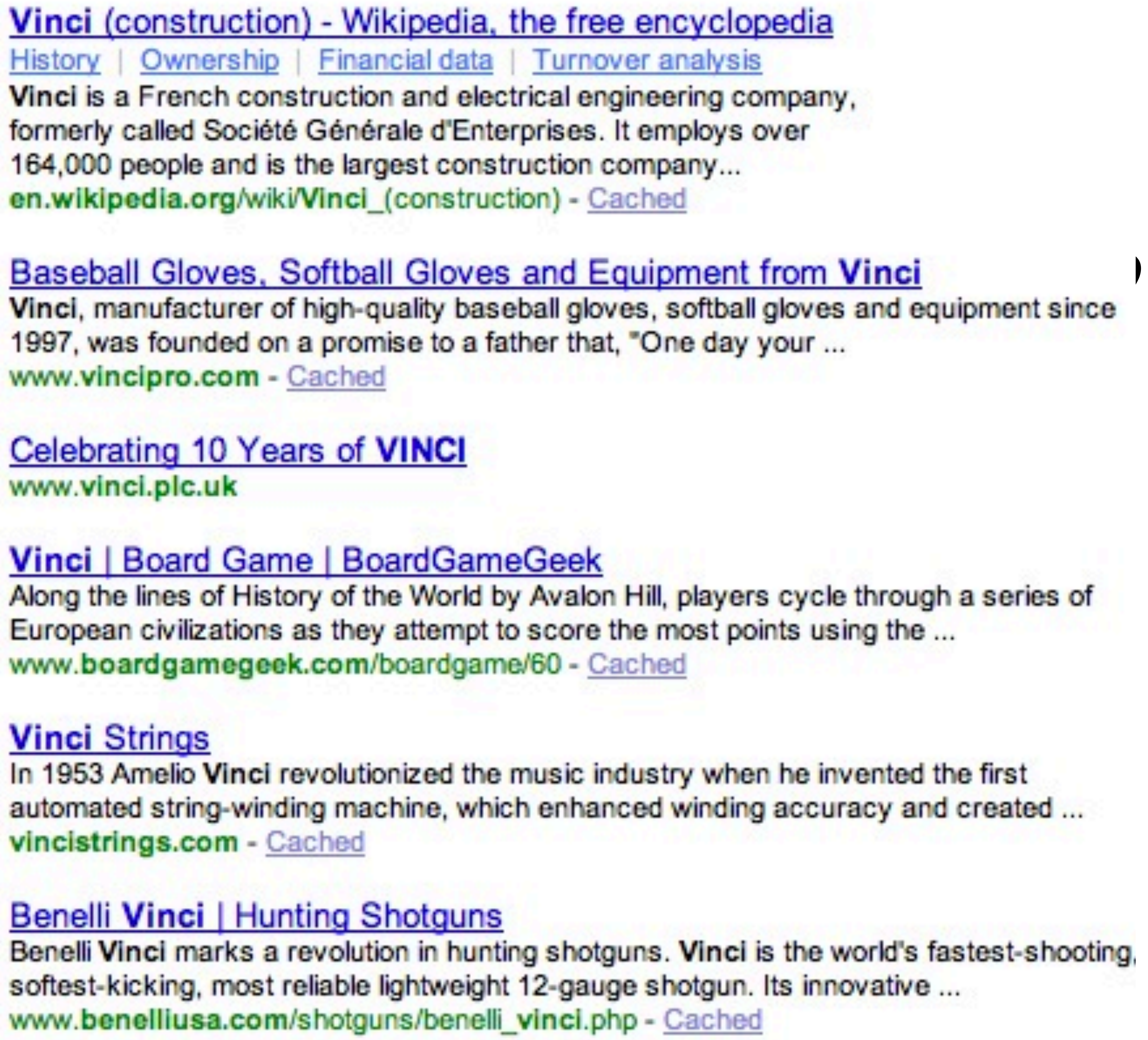


t is the user's intent?

eonar

'inci, t

'inci, t



Results Diversification as a Coverage Problem

- Hypothesis:
  - For each user's **query** I can tell what is the set of all possible **intents**
  - For each **document** in the collection I can tell what are all the possible user's **intents** it represents
    - each **intent** for each **document** is, possibly, **weighted** by a **value** representing how much that intent is represented by that document (e.g., *1/2* of document *D* is related to the intent of "digital photography techniques")
- Goal:
  - Select the set of *k* documents in the collection covering the maximum amount of intent weight. i.e., maximize the number of satisfied users.

State-of-the-Art Methods

- IASelect:
  - Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. **Diversifying search results**. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*, Ricardo Baeza-Yates, Paolo Boldi, Berthier Ribeiro-Neto, and B. Barla Cambazoglu (Eds.). ACM, New York, NY, USA, 5-14.
- xQuAD:
  - Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. **Exploiting query reformulations for Web search result diversification**. In *Proceedings of the 19th International Conference on World Wide Web*, pages 881-890, Raleigh, NC, USA, 2010. ACM.

Diversify(k)

$\textsc{Diversify}(k)$: Given query q, a set of documents R_q, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in \mathcal{D}$ and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$P(S|q) = \sum_c P(c|q) \left(1 - \prod_{d \in S} (1 - V(d|q,c))\right)$$

Diversify(*k*)

intents

DIVERSIFY(k): Given query q, a set of documents R_q, a probability distribution of categories for the query $P\left(c|q\right)$, the quality values of the documents $V(d|q,c)$, $\forall d \in \mathcal{D}$ and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$P\left(S|q\right) = \sum_{c} P\left(c|q\right) \left(1 - \prod_{d \in S} \left(1 - V\left(d|q,c\right)\right)\right)$$

Diversify(*k*)

intents *the weight*

DIVERSIFY(k): Given query q, a set of documents R_q, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in \mathcal{D}$ and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$P(S|q) = \sum_{c} P(c|q) \left(1 - \prod_{d \in S} (1 - V(d|q,c))\right)$$

Diversify(*k*)

intents *the weight*

DIVERSIFY(k): Given query q, a set of documents R_q, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in \mathcal{D}$ and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

is the probability of being relative to intent c

$$P(S|q) = \sum_c P(c|q)\left(1 - \prod_{d \in S}\left(1 - V(d|q,c)\right)\right)$$

Diversify(*k*)

intents

the weight

DIVERSIFY(k): Given query q, a set of documents R_q, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in \mathcal{D}$ and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

is the probability of being relative to intent c

$$P(S|q) = \sum_c P(c|q)\left(1 - \prod_{d \in S}(1 - V(d|q,c))\right)$$

d is not pertinent to c

Diversify(k)

intents

the weight

DIVERSIFY(k): Given query q, a set of documents R_q, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in \mathcal{D}$ and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

is the probability of being relative to intent c

$$P(S|q) = \sum_c P(c|q)\left(1 - \prod_{d \in S}(1 - V(d|q,c))\right)$$

d is not pertinent to c

no doc is pertinent to c

Diversify(k)

Very Large Data Bases 2011

intents

the weight

DIVERSIFY(k): Given query q, a set of documents R_q, a probability distribution of categories for the query $P(c|q)$, the quality values of the documents $V(d|q,c)$, $\forall d \in \mathcal{D}$ and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

is the probability of being relative to intent c

$$P(S|q) = \sum_{c} P(c|q) \left(1 - \prod_{d \in S} (1 - V(d|q,c)) \right)$$

d is not pertinent to c

at least one doc is pertinent to c

no doc is pertinent to c

Known Results

- Diversify(*k*) is NP-hard:
  - Reduction from max-weight coverage
- Diversify(*k*)'s objective function is sub-modular:
  - Admits a *(1-1/e)*-approx. algorithm.
  - The algorithm works by inserting one result at a time, we insert the result with the max marginal utility.
  - Quadratic complexity in the number of results to consider:
    - at each iteration scan the complete list of not-yet-inserted results.

Known Results

- Diversify(*k*) is NP-h
  - Reduction from m

DEFINITION 1 (SUBMODULARITY). *Given a finite ground set N, a set function $f : 2^N \mapsto \mathbb{R}$ is submodular if and only if for all sets $S, T \subseteq N$ such that $S \subseteq T$, and $d \in N \setminus T$, $f(S+d) - f(S) \geq f(T+d) - f(T)$.*

- Diversify(*k*)'s objective function is sub-modular:
  - Admits a *(1-1/e)*-approx. algorithm.
  - The algorithm works by inserting one result at a time, we insert the result with the max marginal utility.
  - Quadratic complexity in the number of results to consider:
    - at each iteration scan the complete list of not-yet-inserted results.

⚠ It looks reasonable, but... ⚠

- ... it may not diversify!
- The objective function is NOT about including as many categories as possible in the final results set.
- It is possible that even if there are less than *k* categories, NOT all categories will be covered:
  - the formulation explicitly considers how well a document satisfies a given category.
- If a category *c* is dominant and not well satisfied, more documents from *c* will be added:
  - possible at the expense of not showing certain categories altogether.

xQuAD_Diversify(*k*)

xQuAD_Diversify(k): Given a query q, a set of ranked documents R_q retrieved for q, a mixing parameter $\lambda \in [0, 1]$, two probability distributions $P(d|q)$ and $P(d, \bar{S}|q)$ measuring, respectively, the likelihood of document d being observed given q, and the likelihood of observing d but not the documents in the solution S. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes for each $d \in S$

$$(1 - \lambda) \cdot P(d|q) \ + \ \lambda \cdot P(d, \bar{S}|q)$$

xQuAD_Diversify(k)

xQuAD_Diversify(k): Given a query q, a set of ranked documents R_q retrieved for q, a mixing parameter $\lambda \in [0, 1]$, two probability distributions $P(d|q)$ and $P(d, \bar{S}|q)$ measuring, respectively, the likelihood of document d being observed given q, and the likelihood of observing d but not the documents i
documents $S \subseteq R_q$ with
$d \in S$

$$P(d, \bar{S}|q) = \sum_{q' \in S_q} \left[P(q'|q) \, P(d|q') \prod_{d_j \in S} 1 - P(d_j|q') \right]$$

$$(1 - \lambda) \cdot P(d|q) \; + \; \lambda \cdot P(d, \bar{S}|q)$$

xQuAD_Diversify(*k*)

xQuAD_Diversify(k): Given a query q, a set of ranked documents R_q retrieved for q, a mixing parameter $\lambda \in [0, 1]$, two probability distributions $P(d|q)$ and $P(d, \bar{S}|q)$ measuring, respectively, the likelihood of document d being observed given q, and the likelihood of observing d but not the documents i... documents $S \subseteq R_q$ with ... $d \in S$

$$P(d, \bar{S}|q) = \sum_{q' \in S_q} \left[P(q'|q)\, P(d|q') \prod_{d_j \in S} 1 - P(d_j|q') \right]$$

$$(1-\lambda) \cdot P(d|q) \; + \; \lambda \cdot P ...$$

Same problem as before...
It may not diversify!

Our Proposal: MaxUtility

Vinci

Our Proposal: MaxUtility

Vinci → Leonardo da Vinci
Vinci → Vinci Town
Vinci → Vinci Group

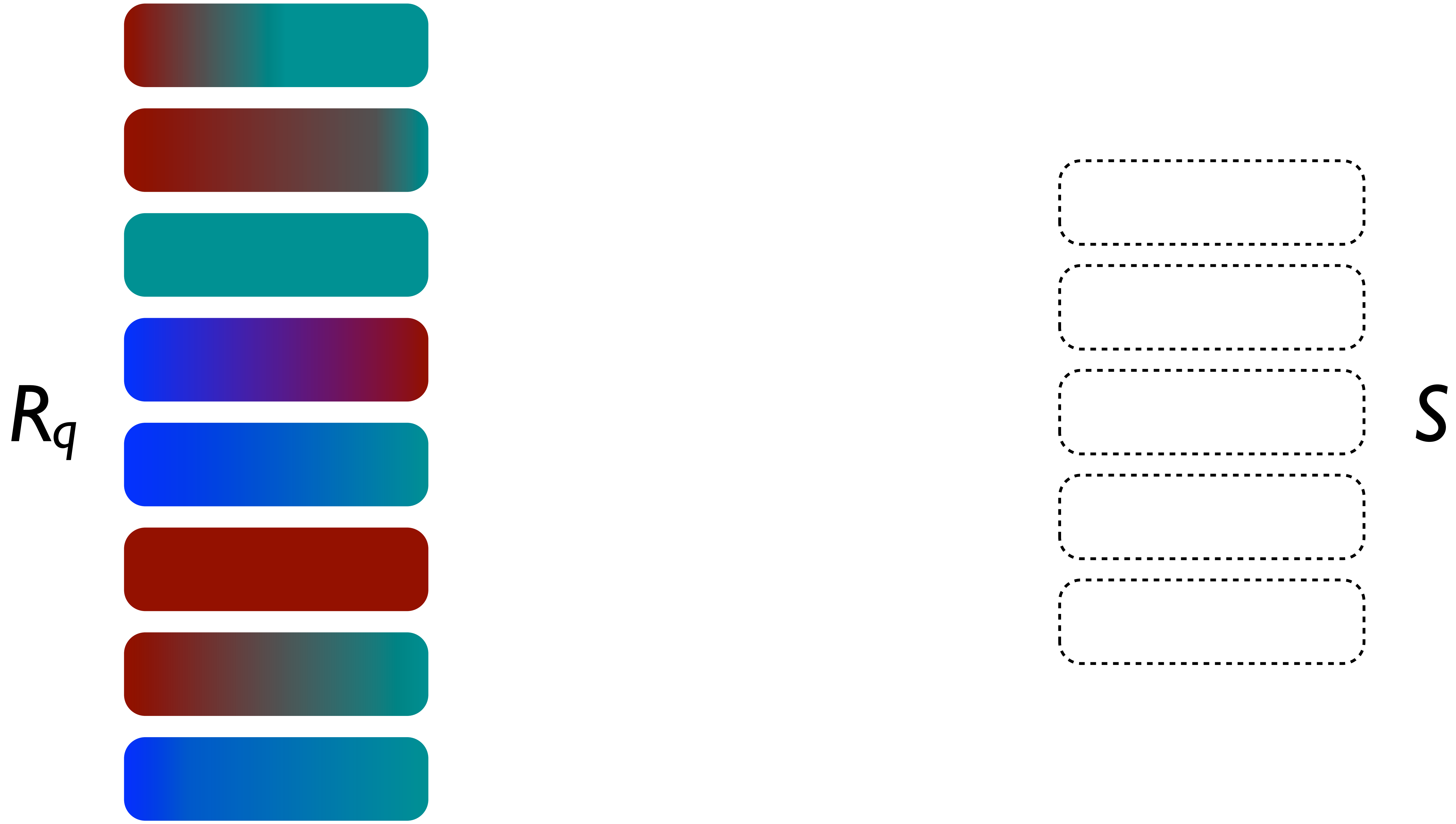
Our Proposal: MaxUtility

Vinci → Leonardo da Vinci 5/12
Vinci → Vinci Town 1/3
Vinci → Vinci Group 1/4

Our Proposal: MaxUtility

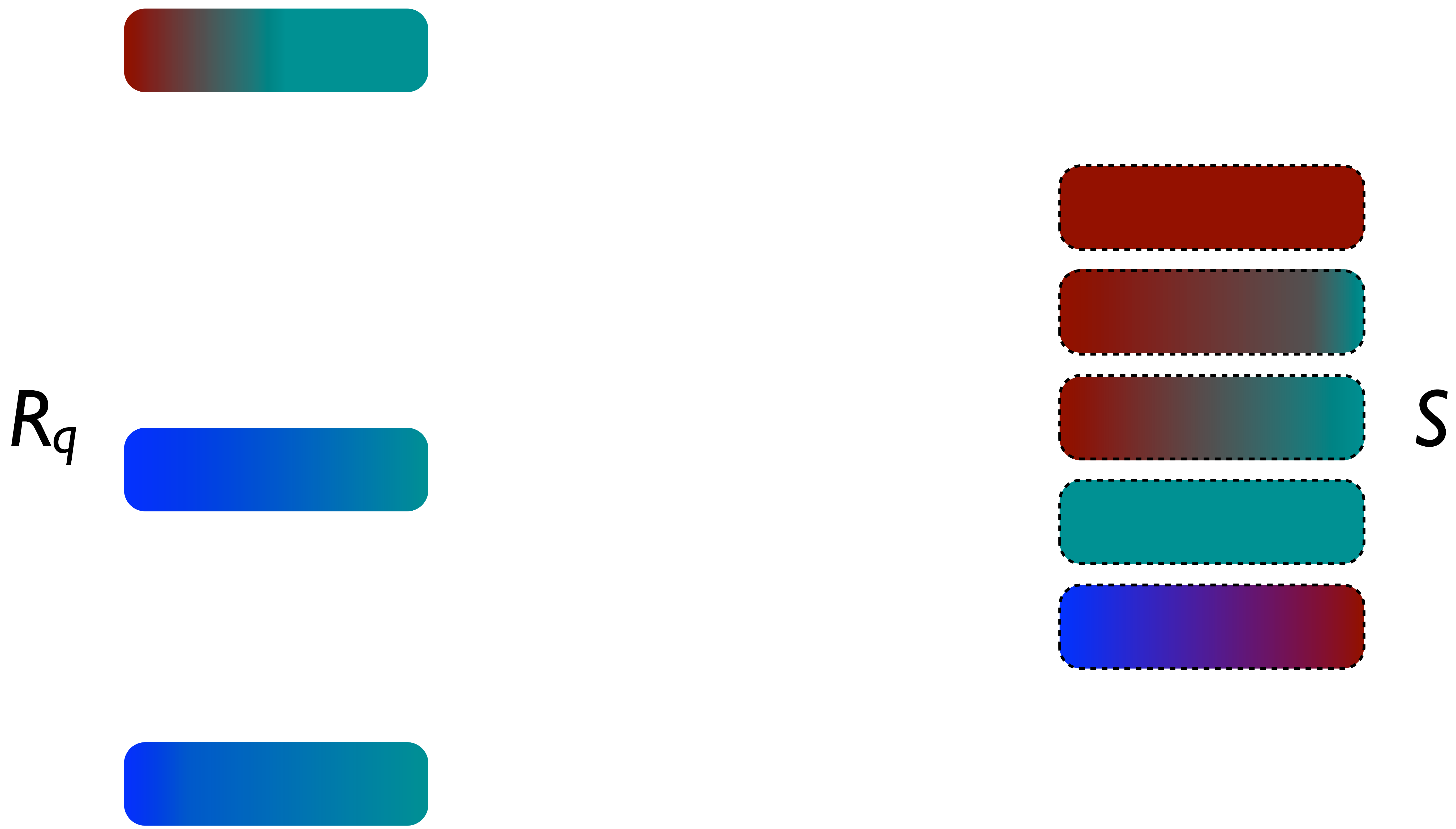
Vinci → Leonardo da Vinci 5/12
Vinci → Vinci Town 1/3
Vinci → Vinci Group 1/4

Our Proposal: MaxUtility



Vinci → Leonardo da Vinci 5/12
Vinci → Vinci Town 1/3
Vinci → Vinci Group 1/4

Our Proposal: MaxUtility



MaxUtility_Diversify(*k*)

MaxUtility_Diversify(k): Given a query q, the set R_q of results for q, two probability distributions $P(d|q)$ and $P\left(q'|q\right) \forall q' \in S_q$ measuring, respectively, the likelihood of document d being observed given q, and the likelihood of having q' as a specialization of q, the utilities $\widetilde{U}(d|R_{q'})$ of documents, a mixing parameter $\lambda \in [0, 1]$, and an integer k. Find a set of documents $S \subseteq R_q$ with $|S| = k$ that maximizes

$$\widetilde{U}\left(S|q\right) = \sum_{d \in S} \sum_{q' \in S_q} (1-\lambda)P(d|q) + \lambda P\left(q'|q\right) \widetilde{U}\left(d|R_{q'}\right)$$

with the constraints that every specialization is covered proportionally to its probability. Formally, let $R_q \bowtie q' = \{d \in R_q | U\left(d|R_{q'}\right) > 0\}$. We require that for each $q' \in S_q$, $|R_q \bowtie q'| \geq \lfloor k \cdot P\left(q'|q\right) \rfloor$.

Why it is Efficient?

- By using a simple arithmetic argument we can show that:

$$\begin{aligned}\widetilde{U}(S|q) &= (1-\lambda)|S_q| \sum_{d \in S} P(d|q) + \\ &+ \lambda \sum_{q' \in S_q} P(q'|q) \sum_{d \in S} \widetilde{U}(d|R_{q'})\end{aligned}$$

- Therefore we can find the optimal set *S* of diversified documents by using a sort-based approach.

OptSelect

Algorithm OptSelect (q, S_q, R_q, k)

$S \leftarrow \emptyset$;
/* Heap(n) instantiates a new n-size heap */
$M \leftarrow$ *new* Heap(k);
For Each $q' \in S_q$ **Do**
    $M_{q'} \leftarrow$ *new* Heap($\lfloor k \cdot P(q'|q) \rfloor + 1$);
    For Each $d \in R_q$ **Do**
        If $\widetilde{U}(d|R_{q'}) > 0$ **Then** $M_{q'}.push(d)$; **Else** $M.push(d)$;
For Each $q' \in S_q$ s.t. $M_{q'} \neq \emptyset$ **Do**
    $x \leftarrow$ pop d with the max $\widetilde{U}(d|q)$ from $M_{q'}$;
    $S \leftarrow S \cup \{x\}$;
While $|S| < k$ **Do**
    $x \leftarrow$ pop d with the max $\widetilde{U}(d|q)$ from M;
    $S \leftarrow S \cup \{x\}$;
Return (S);

OptSelect

Algorithm OptSelect (q, S_q, R_q, k)

```
S ← ∅;
/* Heap(n) instantiates a new n-size heap */
M ← new Heap(k);
For Each q' ∈ S_q Do
    M_q' ← new Heap(⌊k · P(q'|q)⌋ +
    For Each d ∈ R_q Do
        If Ũ(d|R_q') > 0 Then M_q'.pu
For Each q' ∈ S_q s.t. M_q' ≠ ∅ Do
    x ← pop d with the max Ũ(d|q) from M_q';
    S ← S ∪ {x};
While |S| < k Do
    x ← pop d with the max Ũ(d|q) from M;
    S ← S ∪ {x};
Return (S);
```

Algorithm	Complexity
IASelect	$O(nk)$
xQuAD	$O(nk)$
OptSelect	$O(nlog_2k)$

The Specialization Set S_q

- It is crucial for OptSelect to have the set of specialization available for each query.
- Our method is, thus, *query log-based*.
  - we use a query recommender system to obtain a set of queries from which S_q is built by including the most popular (i.e., freq. in query log > *f(q)* / *s*) recommendations:

Algorithm AmbiguousQueryDetect$(q, \mathcal{A}, f(), s)$

/* given the submitted query q, a query recommendation algorithm $\mathcal{A}$, and an integer s compute the set $\widehat{S}_q \subseteq Q$ of possible specializations of q */

1. $\widehat{S}_q \leftarrow \mathcal{A}(q)$;

/* select from $\widehat{S}_q$ the most popular specializations */

2. $S_q \leftarrow \{q' \in \widehat{S}_q | f(q') \geq \frac{f(q)}{s}\}$;
3. **If** $|S_q| \geq 2$ **Then Return** (S_q); **Else Return** $(\emptyset)$;

D. Broccolo, L. Marcon, F.M. Nardini, R. Perego, F. Silvestri
Generating Suggestions for Queries in the Long Tail with an Inverted Index
Information Processing & Management, August 2011

Probability Estimation

$$P(q'|q) = f(q') / \sum_{q' \in S_q} f(q')$$

Usefulness of a Result

Definition (Results' Utility). *The* utility *of a result $d \in R_q$ for a specialization q' is defined as:*

$$U(d|R_{q'}) = \sum_{d' \in R_{q'}} \frac{1 - \delta(d, d')}{rank(d', R_{q'})}.$$

where $R_{q'}$ is the list of results that the search engine returned for specialized query q'.

Usefulness of a Result

DEFINITION (RESULTS' UTILITY). *The* utility *of a result* $d \in R_q$ *for a specialization* q' *is defined as:*

$\delta(d_1, d_2) = 1 - cosine(d_1, d_2)$

$$U(d|R_{q'}) = \sum_{d' \in R_{q'}} \frac{1 - \delta(d, d')}{rank(d', R_{q'})}.$$

where $R_{q'}$ *is the list of results that the search engine returned for specialized query* q'.

Experiments: Settings

- TREC 2009 Web track's Diversity Task framework:
  - ClueWeb-B, the subset of the TREC ClueWeb09 dataset
  - The 50 topics (i.e., queries) provided by TREC
  - We evaluate α-NDCG and IA-P
- All the tests were conducted on a Intel Core 2 Quad PC with 8Gb of RAM and Ubuntu Linux 9.10 (kernel 2.6.31-22).

Experiments: Quality

	c	α-NDCG					IA-P				
		@5	@10	@20	@100	@1000	@5	@10	@20	@100	@1000
DPH Baseline	-	0.190	0.212	0.240	0.303	0.303	0.092	0.093	0.088	0.058	0.006
OptSelect	0	**0.213**	0.227	0.255	0.318	0.352	0.111	0.100	**0.092**	0.061	0.012
	0.05	**0.213**	0.228	0.256	0.319	0.352	**0.112**	**0.101**	0.091	0.061	0.012
	0.10	0.195	0.220	0.246	0.312	0.343	0.102	0.097	0.090	**0.062**	0.012
	0.15	0.190	0.216	0.246	0.305	0.341	0.101	0.098	0.090	0.061	0.012
	0.20	**0.214**	**0.241**	**0.262**	**0.324**	**0.359**	0.110	0.101	0.090	0.060	0.012
	0.25	0.190	0.213	0.238	0.305	0.339	0.095	0.098	0.087	0.058	0.012
	0.35	0.186	0.206	0.235	0.302	0.335	0.089	0.090	0.086	0.058	0.012
	0.50	0.186	0.208	0.236	0.300	0.334	0.091	0.091	0.087	0.058	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	0.093	0.088	0.058	0.012
xQuAD	0	0.211	0.241	0.260	0.320	0.354	0.103	0.102	0.090	0.058	0.012
	0.05	**0.214**	**0.242**	**0.260**	**0.323**	**0.355**	**0.108**	**0.103**	**0.089**	**0.058**	0.012
	0.10	0.193	0.226	0.249	0.308	0.341	0.101	0.101	0.090	0.058	0.012
	0.15	0.200	0.227	0.253	0.315	0.348	0.099	0.095	0.087	0.058	0.012
	0.20	0.204	0.234	0.262	0.321	0.354	0.096	0.099	0.087	0.058	0.012
	0.25	0.181	0.211	0.236	0.303	0.336	0.090	0.095	0.085	0.058	0.012
	0.35	0.185	0.209	0.239	0.302	0.335	0.091	0.092	0.088	0.058	0.012
	0.50	0.190	0.212	0.240	0.303	0.336	0.092	0.093	0.087	0.058	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	0.093	0.088	0.058	0.012
IASelect	0	**0.206**	**0.215**	**0.245**	**0.302**	0.334	0.097	0.089	0.079	0.044	0.009
	0.05	0.205	0.214	0.243	0.299	0.330	**0.098**	0.090	0.078	0.044	0.009
	0.10	0.193	0.200	0.227	0.279	0.309	**0.098**	0.088	0.075	0.039	0.008
	0.15	0.169	0.185	0.207	0.259	0.288	0.089	0.078	0.064	0.039	0.008
	0.20	0.182	0.197	0.229	0.284	0.314	0.085	0.074	0.067	0.046	0.009
	0.25	0.198	0.214	0.243	0.301	0.332	0.092	0.083	0.076	0.052	0.011
	0.35	0.192	0.208	0.241	0.299	0.332	0.095	**0.093**	0.087	0.057	0.012
	0.50	0.192	0.214	0.243	0.306	**0.338**	0.093	0.091	0.087	**0.058**	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	**0.093**	**0.088**	**0.058**	0.012

Experiments: Quality

	c	α-NDCG					IA-P				
		@5	@10	@20	@100	@1000	@5	@10	@20	@100	@1000
DPH Baseline	-	0.190	0.212	0.240	0.303	0.303	0.092	0.093	0.088	0.058	0.006
OptSelect	0	**0.213**	0.227	0.255	0.318	0.352	0.111	0.100	**0.092**	0.061	0.012
	0.05	**0.213**	0.228	0.256	0.319	0.352	**0.112**	**0.101**	0.091	0.061	0.012
	0.10	0.195	0.220	0.246	0.312	0.343	0.102	0.097	0.090	**0.062**	0.012
	0.15	0.190	0.216	0.246	0.305	0.341	0.101	0.098	0.090	0.061	0.012
	0.20	**0.214**	**0.241**	**0.262**	**0.324**	**0.359**	0.110	0.101	0.090	0.060	0.012
	0.25	0.190	0.213	0.238	0.305	0.339	0.095	0.098	0.087	0.058	0.012
	0.35	0.186	0.206	0.235	0.302	0.335	0.089	0.090	0.086	0.058	0.012
	0.50	0.186	0.208	0.236	0.300	0.334	0.091	0.091	0.087	0.058	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	0.093	0.088	0.058	0.012
xQuAD	0	0.211	0.241	0.260	0.320	0.354	0.103	0.102	0.090	0.058	0.012
	0.05	**0.214**	**0.242**	**0.260**	**0.323**	**0.355**	**0.108**	**0.103**	**0.089**	**0.058**	0.012
	0.10	0.193	0.226	0.249	0.308	0.341	0.101	0.101	0.090	0.058	0.012
	0.15	0.200	0.227	0.253	0.315	0.348	0.099	0.095	0.087	0.058	0.012
	0.20	0.204	0.234	0.262	0.321	0.354	0.096	0.099	0.087	0.058	0.012
	0.25	0.181	0.211	0.236	0.303	0.336	0.090	0.095	0.085	0.058	0.012
	0.35	0.185	0.209	0.239	0.302	0.335	0.091	0.092	0.088	0.058	0.012
	0.50	0.190	0.212	0.240	0.303	0.336	0.092	0.093	0.087	0.058	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	0.093	0.088	0.058	0.012
IASelect	0	**0.206**	**0.215**	**0.245**	**0.302**	0.334	0.097	0.089	0.079	0.044	0.009
	0.05	0.205	0.214	0.243	0.299	0.330	**0.098**	0.090	0.078	0.044	0.009
	0.10	0.193	0.200	0.227	0.279	0.309	**0.098**	0.088	0.075	0.039	0.008
	0.15	0.169	0.185	0.207	0.259	0.288	0.089	0.078	0.064	0.039	0.008
	0.20	0.182	0.197	0.229	0.284	0.314	0.085	0.074	0.067	0.046	0.009
	0.25	0.198	0.214	0.243	0.301	0.332	0.092	0.083	0.076	0.052	0.011
	0.35	0.192	0.208	0.241	0.299	0.332	0.095	**0.093**	0.087	0.057	0.012
	0.50	0.192	0.214	0.243	0.306	**0.338**	0.093	0.091	0.087	**0.058**	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	**0.093**	**0.088**	**0.058**	0.012

Experiments: Quality

	c	α-NDCG @5	α-NDCG @10	α-NDCG @20	α-NDCG @100	α-NDCG @1000	IA-P @5	IA-P @10	IA-P @20	IA-P @100	IA-P @1000
DPH Baseline	-	0.190	0.212	0.240	0.303	0.303	0.092	0.093	0.088	0.058	0.006
OptSelect	0	**0.213**	0.227	0.255	0.318	0.352	0.111	0.100	**0.092**	0.061	0.012
	0.05	**0.213**	0.228	0.256	0.319	0.352	**0.112**	**0.101**	0.091	0.061	0.012
	0.10	0.195	0.220	0.246	0.312	0.343	0.102	0.097	0.090	**0.062**	0.012
	0.15	0.190	0.216	0.246	0.305	0.341	0.101	0.098	0.090	0.061	0.012
	0.20	**0.214**	**0.241**	**0.262**	**0.324**	**0.359**	0.110	0.101	0.090	0.060	0.012
	0.25	0.190	0.213	0.238	0.305	0.339	0.095	0.098	0.087	0.058	0.012
	0.35	0.186	0.206	0.235	0.302	0.335	0.089	0.090	0.086	0.058	0.012
	0.50	0.186	0.208	0.236	0.300	0.334	0.091	0.091	0.087	0.058	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	0.093	0.088	0.058	0.012
xQuAD	0	0.211	0.241	0.260	0.320	0.354	0.103	0.102	0.090	0.058	0.012
	0.05	**0.214**	**0.242**	**0.260**	**0.323**	**0.355**	**0.108**	**0.103**	**0.089**	**0.058**	0.012
	0.10	0.193	0.226	0.249	0.308	0.341	0.101	0.101	0.090	0.058	0.012
	0.15	0.200	0.227	0.253	0.315	0.348	0.099	0.095	0.087	0.058	0.012
	0.20	0.204	0.234	0.262	0.321	0.354	0.096	0.099	0.087	0.058	0.012
	0.25	0.181	0.211	0.236	0.303	0.336	0.090	0.095	0.085	0.058	0.012
	0.35	0.185	0.209	0.239	0.302	0.335	0.091	0.092	0.088	0.058	0.012
	0.50	0.190	0.212	0.240	0.303	0.336	0.092	0.093	0.087	0.058	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	0.093	0.088	0.058	0.012
IASelect	0	**0.206**	**0.215**	**0.245**	**0.302**	0.334	0.097	0.089	0.079	0.044	0.009
	0.05	0.205	0.214	0.243	0.299	0.330	**0.098**	0.090	0.078	0.044	0.009
	0.10	0.193	0.200	0.227	0.279	0.309	**0.098**	0.088	0.075	0.039	0.008
	0.15	0.169	0.185	0.207	0.259	0.288	0.089	0.078	0.064	0.039	0.008
	0.20	0.182	0.197	0.229	0.284	0.314	0.085	0.074	0.067	0.046	0.009
	0.25	0.198	0.214	0.243	0.301	0.332	0.092	0.083	0.076	0.052	0.011
	0.35	0.192	0.208	0.241	0.299	0.332	0.095	**0.093**	0.087	0.057	0.012
	0.50	0.192	0.214	0.243	0.306	**0.338**	0.093	0.091	0.087	**0.058**	0.012
	0.75	0.190	0.212	0.240	0.303	0.337	0.092	**0.093**	**0.088**	**0.058**	0.012

Experiments: Efficiency

$\lvert R_q \rvert$	k				
	10	50	100	500	1000
OptSelect					
1,000	0.34	0.58	0.66	0.89	0.98
10,000	1.36	2.13	2.46	3.32	3.57
100,000	4.81	8.32	9.57	12.94	13.92
xQuAD					
1,000	0.43	1.64	3.31	14.82	30.18
10,000	3.27	16.69	32.22	148.41	298.63
100,000	36.27	143.67	285.69	1,425.82	2,849.83
IASelect					
1,000	0.57	1.68	3.92	20.81	39.82
10,000	4.23	23.03	40.82	203.11	409.43
100,000	48.04	205.46	408.61	2,039.22	4,071.81

Conclusions and Future Work

- We studied the problem of search results diversification from an efficiency point of view
- We derived a diversification method (OptSelect):
  - same (or better) quality of the state of the art
  - up to 100 times faster
- Future work:
  - the exploitation of users' search history for personalizing result diversification
  - the use of click-through data to improve our effectiveness results, and
  - the study of a search architecture performing the diversification task in parallel with the document scoring phase (See DDR2011 paper)

Question Time

Franco Maria Nardini
ISTI-CNR, Pisa Italy
http://hpc.isti.cnr.it/~nardini
f.nardini@isti.cnr.it

Backup Slides

α-NDCG

- The α-normalized discounted cumulative gain (α-NDCG) metric balances relevance and diversity through the tuning parameter α.
  - The larger the value of α, the more diversity is rewarded. In contrast, when α = 0, only relevance is rewarded, and this metric is equivalent to the traditional NDCG.
- DCG measures the usefulness, or gain, of a document based on its position in the result list.
  - $$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1+i)}$$
  - Relevance scores might not be binary (i.e., relevant, not relevant) but also indicating how relevant a result is.
  - NDCG is the normalized version of DCG.
- More info at:
  - C. L. Clarke, M. Kolla, G.V. Cormack, O.Vechtomova, A. Ashkan, S. Bu¨ttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In Proc. SIGIR'08, pages 659–666. ACM, 2008.

α-NDCG

- The α-normalized discounted cumulative gain (α-NDCG) metric balances relevance and diversity through the tuning parameter α.
  - The larger the value of α, the more diversity is rewarded. In contrast, when α = 0, only relevance is rewarded, and this metric is equivalent to the traditional NDCG.
- DCG measures the useful
  - $\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1+i}$
  - Relevance scores might
  - NDCG is the normalized version of DCG.
- More info at:
  - C. L. Clarke, M. Kolla, G.V. Cormack, O. Vechtomova, A. Ashkan, S. Bü ttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In Proc. SIGIR'08, pages 659–666. ACM, 2008.

$$\sum_k \frac{1}{\log_2(k+1)} \sum_i r_k^i (1-\alpha)^{s_{i,k-1}} \quad \text{with} \quad s_{i,k-1} := \sum_{j=1}^{k-1} r_j^i$$

IA-P

- Intent Aware - Precision
- As "traditional" precision measured at a certain cutoff
- Basically, precision is weighted on the probability of each intent.
- More info at:
  - Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. **Diversifying search results**. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*, Ricardo Baeza-Yates, Paolo Boldi, Berthier Ribeiro-Neto, and B. Barla Cambazoglu (Eds.). ACM, New York, NY, USA, 5-14.

IA-P

- Intent Aware - Precision
- As “traditional” precision measured at a certain cutoff
- Basically, precision is weighted on

$$\frac{1}{M}\sum_{t=1}^{M}\frac{1}{N_t}\sum_{i=1}^{N_t}\frac{1}{k}\sum_{j=1}^{k} j_t(i,j)$$

- More info at:
  - Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. **Diversifying search results**. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*, Ricardo Baeza-Yates, Paolo Boldi, Berthier Ribeiro-Neto, and B. Barla Cambazoglu (Eds.). ACM, New York, NY, USA, 5-14.